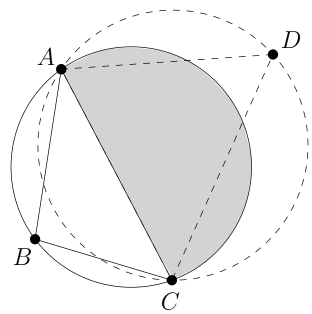
Exercises for CSCI5010

Prepared by Yufei Tao

Problem 1. Let D be a point outside $\odot ABC$ (circumcircle of triangle ABC) such that points B and D fall on different sides of the line passing through segment $\overline{AC}$. Then, $\odot ACD$ covers the territory of arc $\widehat{AC}$ inside $\odot ABC$ (the shadow region in the figure below).



Hint: Pick any point E on the arc $\widehat{AC}$ ($E \neq A$ and $E \neq C$). Prove that the angle $\angle AEC > \angle ADC$.

Problem 2 (reading exercise). Given a triangle ABC and a point p, determine in $O(1)$ time if $\odot ABC$ covers p.

Spoiler: If you do not want to think, read Pg 86 of Prof. Mount's notes.

Problem 3 (Exercise 9.11 from the textbook). Let P be a set of n points in $\mathbb{R}^2$. A Euclidean spanning tree of P is a tree where every vertex is a point in P and every edge is a line segment connecting two points of P. The tree's *weight* equals the total (Euclidean) length of all the segments. The *Euclidean minimum spanning tree* (EMST) is a Euclidean spanning tree of the smallest weight. Give an algorithm to find an EMST of P in $O(n \log n)$ expected time.

Remark: It is $O(n \log n)$ "expected" only because the Delaunay computation algorithm we covered in the lecture is randomized. A Delaunay triangulation can be computed in $O(n \log n)$ worst case time (e.g., by using a planesweep or divide-and-conquer algorithm to compute the corresponding Voronoi diagram).

Problem 4* (Euclidean Traveling Salesman). Let P be a set of n points in $\mathbb{R}^2$. A *tour* is a sequence of n segments $\overline{p_1p_2}, \overline{p_2p_3}, ..., \overline{p_{n-1}p_n}, \overline{p_np_1}$, where each p_i ($i \in [1, n]$) is a distinct point in P. The *length* of the tour is the total length of all the n segments. Let ℓ^* be the shortest length of all possible tours. Design an algorithm to find a tour with length at most $2\ell^*$ in $O(n \log n)$ expected time.

Hint: Visit Prof. Tao's CSCI3160 website (`https://www.cse.cuhk.edu.hk/~taoyf/course/3160/23-fall`) and look for "Traveling Salesman".

Problem 5* (Clustering; textbook exercise 9.16). Let P be a set of n points in $\mathbb{R}^2$. A *k-clustering* of P is a partition $P_1, P_2, ..., P_k$ of P such that

- each P_i ($i \in [1, k]$) is a non-empty subset of P,
- $P_i \cap P_j = \emptyset$ for any different $i, j \in [1, k]$, and

- $P_1 \cup P_2 \cup ... \cup P_k = P$.

We will refer to each P_i ($i \in [1, k]$) as a *cluster*. For any different $i, j \in [i, k]$, define the *distance* between clusters P_i and P_j as

$$dist(P_i, P_j) \quad = \quad \min_{p \in P_i, q \in P_j} dist(p, q)$$

where $dist(p, q)$ is the Euclidean distance between points p and q. The *quality* of the k-clustering is defined to be the smallest distance between all $\binom{k}{2}$ cluster pairs.

- Prove: If the quality of P is determined by two points $p, q \in P$, then $\{p, q\}$ is an edge in the Delaunay triangulation of P.
- Given a real value $r > 0$ and an integer $k \geq 2$, give an algorithm to determine whether there exists a k-clustering of P whose quality is at least r. Your algorithm needs to finish in $O(n \log n)$ expected time.

Hint: The first question is easy. For the second question, what happens if we remove all the edges of the Delaunay graph of P that have lengths greater than r?